

Erzeugen von 32 Bit Konstanten:

Wichtig: Das Register besteht aus 32 Bit

slli (shift logical left immediate)

1. lwi (load upper immediate): lädt eine Konstante (immediate) in das Register und shiftet den Wert um 12 Bit nach links \Rightarrow die oberen 20 Bit beinhalten die Zahl und die unteren 12 Bit werden 0

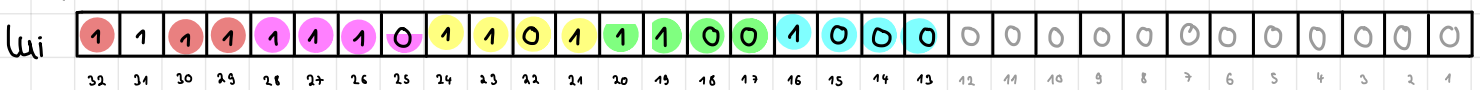
Aufgabe: lade folgenden 32 bit Wert in ein Register:

$$0x \text{ FEDC8765}_{\text{hex}} = \underbrace{1111}_F \cdot \underbrace{1110}_E \cdot \underbrace{1101}_D \cdot \underbrace{1100}_C \cdot \underbrace{1000}_8 \cdot \underbrace{0111}_7 \cdot \underbrace{0110}_6 \cdot \underbrace{0101}_5_{\text{bin}} = 4\ 275\ 863\ 445_{\text{dec}}$$

Unser Register

lwi 50, 0x FEDC8

slli (shift left logical immediate)



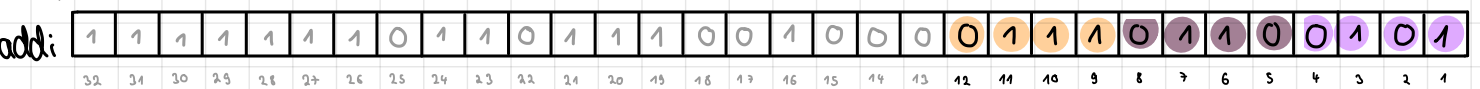
\Rightarrow In unser Register wurde bis jetzt nur der Wert $0x \text{ FEDC8000}$ geladen (die oberen 20 Bit)

\rightarrow Es fehlt noch der Wert $0x 765$ (die unteren 12 Bit)

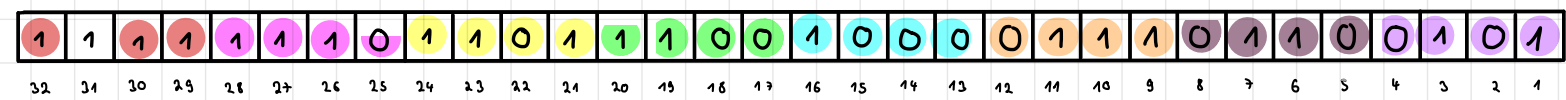
2. addi (add immediate) addiert die unteren 12 Bit hinzu

Diese 12 bit werden mit dem addi Befehl zu unserem bestehenden Register hinzugefügt:

addi 50, 0x 765



Resultat:



Bei addi von 12 Bit Immediate Werten mit Vorzeichenerweiterung (sign extension) kann es zu Problemen kommen

Wichtig: $\rightarrow 255$ vs -1

12 bit Konstanten (0 bis $2^{12}-1 = 4095$) können mit addi in einem Register gespeichert werden.

32 bit Konstanten (0 bis $2^{32}-1 = 4294967295$) müssen in zwei Schritte aufgeteilt werden. Zuerst die ersten

20 Bit (5 hex Stellen) mit lwi in ein Register laden. Danach die fehlenden 12 bit (3 hex Stellen) mit addi

zum bestehenden Register hinzufügen. ($20 \text{ Bit} + 12 \text{ Bit} = 32 \text{ Bit}$).